Bootcamp – 1: Linux do Zero

Aluno: Francisco Castro Del`Gaudio Junior

1 ano de Ciência da Computação

**PREPARE-SE PARA A JORNADA – CURSO: SEJA PROTAGONISTA NESTE BOOTCAMP**

1. **Apresentação e objetivo geral:**

O percurso se dá em 4 partes: Parte 1 – Fala da metodologia de ensino DIO, organização dos módulos, desafios etc. 2 – Como resolver os desafios de código; 3 – Desafios de projeto com responsabilidade diferente e; 4- A conclusão dos ensinos.

Metodologia de ensino do Bootcamp: Cursos – fundamentação teórica e conceitos essenciais, Mentorias – interação para tirar dúvidas e emergir nos ensinos do curso, Desafios de Código – promove a logica e pensamento computacional. (o dia a dia do programador), Desafios de Projeto - traz a pratica e a criação de portfólio para entrevistas técnicas e mercado de trabalho.

Desafio de projeto – Todos são feitos utilizando repositórios no github com linguagem git. Uma boa dica na hora de criar nosso repositório e dar “fork” onde ele vai trazer todo o código original para o seu repositório, lincando ele, e mostrando suas alterações.

**PREPARE-SE PARA A JORNADA – CURSO: INTRODUCAO AO DESENVOVIMENTO MODERNO DE SOFTWARE**

Na criação de software pensamos na solução do problema e quais ferramentas serão utilizadas para solucionar.

Sistema Desktop x Sistema Web – Sistemas Desktop são pensados num programa de arquivo executável no computador, como o Excel, jogos etc. Porém existe a desvantagem da falta de mobilidade do arquivo. Os Sistemas Web podem ser utilizados remotamente sem necessidade de deslocar nenhuma máquina física.

UX/UI Designer – UX Design (User Experience), e um processo para melhorar a usabilidade, a acessibilidade e a satisfação na interação. Eles montam o esqueleto de como será realizado a interação. UI Design (User Interface) pode trabalhar com a mesma coisa que o UX, mas e focado em cores, fontes de letras, cores etc.

Front End – Responsável pela aplicação do requerente de dados, trabalha com a parte visual e interativa. As tecnologias base são HTML, CSS e JavaScript. Modelo Cliente-Servidor – Existe o fornecedor de dados e os requerentes, e o front end e responsável pela aplicação do lado do requerente.

Framework – E um facilitador, traz soluções já pré-definidas, eles resolvem a parte repetitiva e mecânica de um programador, que não exige a solução de problemas.

Back End – Responsável pela parte do servidor, de tudo que existe por trás da aplicação. Uma API – E o caminho entre um aplicativo e outro, a interação entre o front end e um banco de dados, por exemplo.

QA – (Quality Assurance), cuida da qualidade do software, como bugs, segurança etc. Não necessariamente conhece de programação, mas de testes e criação de relatórios.

Infraestrutura – Diz respeito da locação física de espaço dos servidores, servidores privados, públicos, locais etc. Um data center. Diz respeito também a redes, cabeamento, HDs etc.

Cloud – Nuvem publica, empresas que alocam os dados nos servidores deles. Profissional de Cloud Computing resolve o sistema que será utilizado, a integração das componentes e etc.

**PREPARE-SE PARA A JORNADA – CURSO: TRABALHANDO EM EQUIPES AGEIS**

## O que é metodologia ágil? Uma metodologia ágil é um conceito de gerenciamento de projetos que promove mais dinamismo ao processo. Gestores conseguem ter uma visão mais simples e prática das tarefas realizadas pela sua equipe, deixando de lado os processos lentos de entrega, otimizando todas as etapas produtivas.Valorizar mais os indivíduos que processos. O software funcionando mais do que documentação abrangente. Se adaptar e melhor do que seguir o plano à risca. Colaborar com o cliente, mais do que negociações.

## Processo: Recorrente e contínuo, geralmente repetitivo. Projeto: Esforço temporário, resultado exclusivo e tem início e fim bem definidos. O que e gestão ágil? E um tipo de gestão de projetos, flexível, colaborativa, voltada ao aprendizado e a entrega continua.

Métodos e ferramentas Ágeis ->

Framework Scrum – As bases do Scrum são o empirismo (observação e experiencia) e Lean Thinking (Ser objetivo, sem desperdícios como o tempo). Pilares do Scrum: Transparência, inspeção e adaptação são os pilares do Scrum. O Scrum não detalha as instruções, mas orienta as interações.

Diagrama

Descrição gerada automaticamente

Método XP – Extreme Programming, criada com foco no desenvolvimento de sistemas, interação com os clientes, testagem constante e ciclos curtos de desenvolvimento. Objetivo ciclos de entregas rápidos, contínuos e incrementais.

Kanban – Definir, gerencia e melhorar serviços. Criado pela Toyota para gestão de estoque e desperdício, para controle de estoque de matérias. O desperdício e o grande inimigo do Kanban. TODO, Doing, Done. Colunas que indicam o status da tarefa.

OKR – Objetivo e simplificar a forma de encarar os principais objetivos estratégicos de uma empresa. Claro, inspirador, desafiador e alinhado com a missão do negócio. Os Key Results são as metas que determinam o atingimento do objetivo na metodologia OKR.

Exemplo de OKR:

Objetivo: Criar uma experiencia inesquecível para o cliente em todos os pontos de contato.

Key Results: Integrar a base de dados dos 5 canais de atendimento; contratar mais 15 atendentes e 1 gestor de comunidade; alcançar um NPS de 80%.

Benefícios: Agilidade, cooperação, transparência, monitoramento contínuo.

Ferramentas para equipes ágeis: Uma imagem contendo Diagrama

Descrição gerada automaticamente

**PREPARE-SE PARA A JORNADA – CURSO: INTRODUCAO A PROGRAMACAO E PENSAMENTO COMPUTACIONAL**

**Muito importante: Pensamento Computacional**

Decomposição – Dividir um problema difícil em problemas menores e mais fáceis. Ex: Bolo e o problema difícil, mas se pensar que um bolo e feito de massa, recheio e cobertura, e resolvo um problema de cada vez, se torna mais fácil e o bolo está decomposto.

Reconhecimento de padrões – Similaridade e diferenças entre os problemas para encontrar padrões e tirar conclusões a partir disso.

Abstração – Extrapolar o conceito do problema para uma forma generalista, tirar do particular para o generalista. Ex: Se eu consigo descobrir a circunferência de um círculo qualquer, eu consigo descobrir qualquer circunferência.

Design de algoritmos – Definir passo a passo a solução do problema. Definir uma solução; testar a solução; aperfeiçoar a solução encontrada.

Habilidades Complementares – Raciocínio logico e aperfeiçoamento. Raciocínio logico e um pensamento estruturado que permite encontrar a conclusão ou determinar a resolução do problema.

Raciocínio logico: Indução, dedução e abdução.

Introdução a lógica de programação – Serve para resolver um problema. Organização e planejamento das instruções, assertivas em um algoritmo, a fim de viabilizar a implantação de um programa.

Técnicas de logica de programação – Técnica Linear: Organização de elementos por uma única propriedade.

Técnica Estruturada: Escrita, entendimento, validação e manutenção.

Técnica Modular: Simplificação, decomposição e verificação do modulo.

Estruturas de Repetição – Enquanto: Enquanto algo esteja acontecendo ele vai repetir as instruções. Repita, Para.

**PREPARE-SE PARA A JORNADA – CURSO: GIT/GITHUB**

Git clone – comando para clonar tudo que tem no repositório na pasta.